

Tree Model Guided (TMG) Enumeration as the Basis for Mining Frequent Patterns from XML Documents

Henry Tan Setiawan

9/28/2007

A Thesis presented for the degree of Doctor of Philosophy

Faculty of Information Technology
University of Technology, Sydney
Australia

Curriculum Vitae

Henry Tan was born in a small town, Sukabumi, Indonesia, on December 7th, 1979. He obtained his Bachelor of Computer System Engineering with first class honour from La Trobe University, VIC, Australia in 2003. During his undergraduate study, he was nominated as the most outstanding Honours Student in Computer Science. Additionally, he was the holder of 2003 ACS Student Award. After he finished his Honour year at La Trobe University, on August 2003, he continued his study pursuing his doctorate degree at UTS under supervision Prof. Tharam S. Dillon. His research interests include Data Mining, Computer Graphics, Game Programming, Neural Network, AI, and Software Development. On January 2006 he took the job offer from Microsoft Redmond, USA as a Software Design Engineer (SDE).

Dedications

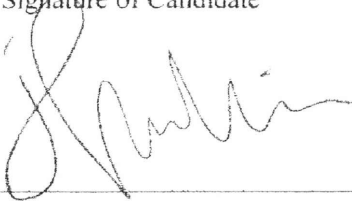
My study and this thesis is about learning and presenting the truth. Therefore, first I would like to dedicate the thesis to the $\lambda\omicron\gamma\omicron\varsigma$ through which the truth is manifested. I thank my wife Theresia Liu for consistently supporting me during the hard work and struggles of this thesis. Last but not least, I devote the thesis to my parents, Tan Djoe Jin and Toe Oey Jin who have been instrumental in guiding my life and encouraging me to succeed.

Certificate of Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

A handwritten signature in black ink, written over a horizontal line. The signature is cursive and appears to be 'J. Smith'.

Acknowledgements

I would like to acknowledge a number of inspirational people for all the help, assistance, support and encouragement I have received during my PhD candidature.

Firstly and most especially, I would like to thank Prof. Tharam S. Dillon who has been the great mentor, teacher, and supervisor throughout my research life since my early years at La Trobe University completing my Bachelor of Computer Systems Engineering, Honours. Your guidance on work related issues, as well as your wise advice on matters of life and career, went beyond my expectations of a supervisor. Your time, attention and dedication to the research works carried out throughout my PhD have been outstanding and invaluable.

I would like to thank my wife who has been by my side throughout all the struggles and up and down cycles of my PhD research, also for presenting me with my cherished and much loved daughter, Enrica Tan, who was the main reason that this PhD thesis is completed. I would like to thank my parents who have given me all the necessary financial support, advice and moral guidance throughout my life and my study. Furthermore, I would like to thank my great colleague, Fedja Hadzic. He has been a great partner with a great, positive attitude and dedication through whom most of the important works in this thesis were conceived and published. Thank you to all of my colleagues from the Exel Research group for sharing great times. Last but not least, I would like to thank Dr. Ling Feng for her feedback during the initial stages of my study. Special thanks to Dr. Yun Chi who has always been very helpful providing the source code and related materials. Thanks also to Dr. Mohammed Zaki who has been very helpful providing the source code and answering a lot of questions throughout our discussions.

List of Publications

1. Tan, H, Dillon, TS, Feng, L, Chang, E & Hadzic, F 2005, 'X3-Miner: Mining patterns from XML database', in A Zanasi, CA Brebbia & NFF Ebecken (eds), *Proceedings of the 6th International Conference on Data Mining (Data Mining'05)*, Skiathos, Greece, WIT Press, pp. 287-297.
2. Tan, H, Dillon, TS, Hadzic, F, Feng, L & Chang, E 2005, 'MB3-Miner: Mining eMBedded subTREEs using tree model guided candidate generation', *Proceedings of the 1st International Workshop on Mining Complex Data (MCD'05)*, Houston, TX, USA, pp. 103-110.
3. Tan, H, Dillon, TS, Hadzic, F, Chang, E & Feng, L 2006, 'IMB3-Miner: Mining induced/embedded subtrees by constraining the level of embedding', In WK Ng, M Kitsuregawa & J Li (eds), *Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'06)*, Singapore, pp. 450-461.
4. Tan, H, Hadzic, F, Dillon, TS & Chang, E 2008, 'State of the art of data mining of tree structured information', *Computer System Science and Engineering*, vol. 23, no. 4, July 2008 (pending publication).
5. Tan, H, Dillon, TS & Hadzic, F 2006, 'Razor: Distance constrained mining of embedded subtrees', in Tsumota & Shusaku (eds), *Proceedings of the International Conference on Data Mining (ICDM'06)*, Hongkong, pp. 8-13.
6. Tan, H, Dillon, TS, Hadzic, F, Feng, L & Chang, E 2007, 'Tree model guided candidate generation for mining frequent subtrees from XML', Publication pending in *Transactions on Knowledge Discovery from Data (TKDD)*.
7. Tan, H, Dillon, TS, Hadzic, F, Chang, E & Feng, L 2007, 'Mining induced /embedded subtrees using the level of embedding constraint', submitted to *Fundamenta Informaticae*.

8. Tan, H, Dillon, TS, Hadzic, F & Chang, E 2006, 'SEQUEST: Mining frequent subsequences using DMA strips', in A Zanasi, CA Brebbia & NFF Ebecken (eds), *Proceedings of the 7th International Conference on Data Mining and Information Engineering (Data Mining'06)*, Prague, Czech Republic, WIT Press, pp. 315-328.
9. Hadzic, F, Dillon, TS, Sidhu, AS, Chang, E & Tan, H 2006, 'Mining substructures in protein data', *Proceedings of the 6th International Conference on Data Mining Workshop (ICDMW'06) - Invited*, Hong Kong, pp. 213-217.
10. Hadzic, F, Tan, H & Dillon, TS 2007, 'UNI3 - efficient algorithm for mining unordered induced subtrees using TMG candidate generation', *Proceedings of the Computational Intelligence and Data Mining (CIDM'07)*, Hawaii, USA, pp. 568-575.
11. Hadzic, F, Tan, H, Dillon, TS & Chang, E 2007, 'U3: Unordered subtree mining using TMG candidate generation and the level of embedding constraint', submitted to ECML PKDD 2008.
12. Hadzic, F, Tan, H, Dillon, TS & Chang, E 2007, 'Implications of frequent subtree mining using hybrid support definition', in A Zanasi, CA Brebbia & NFF Ebecken (eds), *Proceedings of the 8th International Conference on Data Mining & Information Engineering (Data Mining'07)*, The New Forest, UK, WIT Press, pp. 13-24.
13. Hadzic, F, Dillon, TS & Tan, H 2007, 'Outlier detection strategy using the self-organizing map', in X Zhu & I Davidson (eds), *Knowledge Discovery and Data Mining: Challenges and Realities*, Information Science Reference, Hershey, PA, USA, pp. 224-243.
14. Hadzic, F, Dillon, TS, Tan, H, Feng, L & Chang, E 2007, 'Mining frequent patterns using self-organizing map', in D Taniar (ed.), *Research and Trends in*

Data Mining Technologies and Applications: Advances in Data Warehousing and Mining, IGI Global, Hershey, PA, USA, pp. 121-135.

15. Sidhu, AS, Dillon, TS & Setiawan, H 2004, 'XML based semantic protein map', in A Zanasi, NFF Ebecken & CA Brebbia (eds), *Proceedings of 5th International Conference on Data Mining, Text Mining and their Business Applications (Data Mining '04)*, Malaga, Spain, WIT Press, pp. 51-60.
16. Sidhu, AS, Dillon, TS & Setiawan, H 2004, 'Comprehensive protein database representation', in A Gramada & PE Bourne (eds), *Proceedings of the 8th International Conference on Research in Computational Biology (RECOMB '04)*, ACM Press, San Diego, CA, USA, pp. 427-429.
17. Sidhu, AS, Dillon, TS, Sidhu, BS & Setiawan, H 2004, 'Protein knowledge meta model', *Molecular & Cellular Proteomics*, pp. 262-263.

Table of Contents

Curriculum Vitae	ii
Dedications	iii
Certificate of Authorship	iv
Acknowledgements	v
List of Publications	vi
Table of Contents	ix
Table of Figures	xviii
Abstract	xxi
Chapter 1 Introduction	1
1.1 Motivating Examples	3
1.1.1 Affinity Grouping	4
1.1.2 Fraud Detection	4
1.1.3 Genomic Data	5
1.1.4 Text Mining	5
1.1.5 Pattern Detection	6
1.2 Data Mining as a Phase of Knowledge Discovery	6
1.3 Data Mining Tasks	8
1.3.1 Predictive Tasks	8
1.3.1.1 Classification	8
1.3.1.2 Estimation	9
1.3.1.3 Outlier Analysis	9
1.3.2 Descriptive Tasks	9
1.3.2.1 Association Analysis	9
1.3.2.2 Clustering	10
1.3.2.3 Sequential Pattern Discovery	10
1.4 Type of Data	11
1.4.1 Structured Data	11

1.4.2 Semi-structured Data	11
1.4.3 Unstructured Data	12
1.5 Type of Patterns in Data	12
1.5.1 Sequential Pattern	12
1.5.2 Tree-structured Pattern	12
1.5.3 Graph-structured Pattern	13
1.6 Motivations for the Thesis	13
1.7 Scope of the Thesis	18
1.8 Plan of the Thesis	19
References	24
Chapter 2 Existing Works	31
2.1 Introduction	31
2.2 Association Mining (General)	32
2.2.1 Elements of Association Mining	32
2.2.1.1 Frequent Pattern Mining	33
2.2.1.2 Association Rule Generation	34
2.2.2 Problem of Candidate Enumeration	35
2.2.3 Problem of Frequency Counting	36
2.3 Mining Frequent Subtrees	37
2.3.1 XML is an Emerging Source of Semi-structured Data	38
2.3.2 Challenges of Mining More Complex Data	39
2.3.3 Subtree Encoding (String Encoding)	40
2.3.4 Different Types of Subtree	41
2.3.5 Constraints	42
2.3.6 Issues in Enumeration on Tree-structured Data	43
2.3.6.1 Enumeration Strategies	43
2.3.6.2 Enumeration by Join	43
2.3.6.3 Structure-guided Enumeration	45
2.3.6.4 Enumeration by Extension	46
2.3.6.5 Horizontal and Vertical Enumeration	47
2.3.7 Issues in Frequency Counting on Tree-structured Data	48
2.3.7.1 Transaction-based Support	49
2.3.7.2 Occurrence-match (Weighted) Support	49

2.3.8 Mining Frequent Unordered & Ordered Subtrees	49
2.4 Mining Frequent Subsequences	51
2.5 Conclusions	53
References	56
Chapter 3 General Concepts, Definitions and Problem Definitions	72
3.1 Introduction	72
3.2 General Tree Concepts and Definitions	72
3.2.1 Uniform Tree	73
3.2.2 Bijection	74
3.2.3 Tree Isomorphism	74
3.2.4 Ordered Tree	74
3.2.5 Unordered Tree	74
3.2.6 Right-most-path (RMP)	74
3.2.7 Different Types of Subtree	75
3.2.7.1 Induced Subtree	75
3.2.7.2 Embedded Subtree	75
3.3 Problem of Association Rule Mining	76
3.3.1 Association Rule Framework	76
3.3.1.1 Support	76
3.3.1.2 Confidence	77
3.3.1.3 Frequent Itemset Discovery	78
3.3.1.4 Rule Generation	79
3.4 XML and Association Mining	79
3.4.1 XML Document Entities	80
3.4.1.1 XML Nodes	80
3.4.1.2 Element-attribute Relationships	81
3.4.1.3 Element-element Relationships	81
3.4.1.4 Tree-structured Items	81
3.4.2 Parallelism between XML and Tree Structure	82
3.4.3 Problem of XML Document Association Mining	83
3.5 Problem of Mining Frequent Subtrees from a Database of Trees	83
3.5.1 Induced versus Embedded Subtree	84
3.5.2 Candidate Generation	84

3.5.3 Support Definitions	85
3.5.3.1 Transaction-based Support	85
3.5.3.2 Occurrence-match Support	85
3.5.4 Frequency Counting	87
3.5.5 Constraints	87
3.5.5.1 Level of Embedding and Maximum Level of Embedding Constraint	87
3.5.5.2 Distance Constraint	88
3.5.6 Transaction-based versus Occurrence-match Support	89
3.5.7 Issues and Concepts of Mining Frequent Unordered Subtree	91
3.6 General Sequence Concepts and Definitions	96
3.6.1.1 Problem of Mining Frequent Subsequences from a Database of Sequences	98
3.7 Summary	98
References	99
<i>Chapter 4 Overview of the Proposed Framework</i>	105
4.1 Introduction	105
4.2 The Goals of the Development	105
4.3 Representations & Data Structures	107
4.3.1 Efficient Canonical Form Representation	108
4.3.1.1 Canonical Representations for Labeled Trees	108
4.3.1.2 Canonical Representations for Unordered Trees	109
4.3.2 Efficient Data Structures for Trees	109
4.3.2.1 Dictionary	110
4.3.2.2 Embedding List	110
4.3.2.3 Vertical Occurrence List (VOL)	111
4.3.2.4 Recursive List (RL)	111
4.3.2.5 RMP Coordinate List	111
4.4 Mining Frequent Ordered Subtrees	111
4.4.1 Constraint-driven Approach	113
4.4.1.1 Feasible Computation through Maximum Level of Embedding Constraint	113
4.4.1.2 Splitting Embedded Subtree through Distance Constraint	114
4.4.2 Tree Model Guided (TMG) Enumeration	116
4.4.3 Frequency Counting with Vertical Occurrence List	117
4.4.3.1 Issue of using Occurrence-match Support	118

4.4.3.2 Full (k-1) Pruning as Solution to Pseudo-frequent Subtree Problem	120
4.4.4 Optimizations Techniques	120
4.4.4.1 Utilization of RMP Coordinate List	120
4.4.4.2 Utilization of Recursive List	121
4.5 Mining Frequent Unordered Subtrees	122
4.6 Mining Frequent Sequential Patterns using Vertical Tree Approach	123
4.7 Summary	124
References	126
<i>Chapter 5 Representations and Data Structures</i>	129
5.1 Introduction	129
5.2 Efficient Canonical Form Representation	131
5.2.1 Canonical Representations for Labeled Trees	131
5.2.2 Canonical Representations for Unordered Trees	133
5.2.2.1 Breadth-first Canonical Form (BFCF) String Encoding	133
5.2.2.2 Depth-first Canonical Form (DFCF) String Encoding	134
5.2.3 Efficient Representation for Processing XML Documents	135
5.3 Efficient Representations and Data Structures for Trees	138
5.3.1 Dictionary	139
5.3.2 Embedding List (EL)	141
5.3.3 Vertical Occurrence List (VOL)	143
5.3.4 Recursive List (RL)	145
5.3.5 RMP Coordinate List	146
5.4 Summary	149
References	150
<i>Chapter 6 Mining Frequent Ordered Subtrees</i>	152
6.1 Introduction	152
6.2 Introduction to the Framework	153
6.3 General Steps of the Proposed Framework for Mining Frequent Subtrees	154
6.4 MB3 Miner: Mining Frequent Ordered Embedded Subtrees	157
6.4.1 Generating the Dictionary (Module C) & Frequent 1-subtrees (F_1)	158

6.4.2 Generating an Embedding List (EL) (Module D) & Frequent 2-subtrees (F_2)	159
6.4.3 Generate k-subtrees (C_k) (Module E)	160
6.4.3.1 Occurrence Coordinate	160
6.4.3.2 The Right-most-path (RMP) Coordinate	161
6.4.3.3 Extension Coordinates	161
6.4.3.4 Extension Point	162
6.4.3.5 Enumerating Embedded Subtrees using the TMG Enumeration Technique	162
6.4.3.6 Step-wise Depth-first String Encoding Generation	163
6.4.4 Frequency Counting	164
6.4.4.1 Full (k-1) Pruning	166
6.4.4.2 Accelerating Full Pruning	167
6.4.5 MB3 Miner Pseudo-code	168
6.5 iMB3 Miner: Mining Frequent Ordered Induced Subtrees	168
6.5.1 Utilization of the Maximum Level of Embedding Constraint	170
6.5.2 Enumerating Induced/Embedded Subtrees using TMG Enumeration (Module F)	171
6.5.3 iMB3 Miner Pseudo-code	172
6.6 RAZOR: Mining Distance-constrained Embedded Subtrees	172
6.6.1 Motivating Examples	173
6.6.2 Definition of a Distance-constrained Embedded Subtree	176
6.6.3 Illustration of Distance Constraint	176
6.6.4 Generating Distance-constrained String Encoding	177
6.7 [i]MB3^R Miner: Optimization through Utilization of the Recursive List and the RMP Coordinate	178
6.7.1 Space Optimization Utilizing the Recursive List (RL)	178
6.7.2 Generate k-subtrees (C_k) (Module E)	179
6.7.2.1 Enumerating Induced/Embedded Subtrees with RMP Optimization (Module F)	180
6.7.3 Frequency Counting Optimization Utilizing the RMP Coordinate List	181
6.7.4 [i]MB3 ^R Miner Pseudo-code	183
6.8 Summary	184
References	187
Chapter 7 Mathematical Analysis	190
7.1 Introduction	190
7.2 Mathematical Model of the TMG Enumeration Approach	190

7.2.1 Complexity of 1-subtree Enumeration	192
7.2.2 Complexity of 2-subtree Enumeration	192
7.2.3 Complexity of k-subtree Enumeration	194
7.2.4 Analyzing TMG Enumeration Cost Graph	197
7.2.5 Overall Remarks	199
7.3 Complexity Analysis of Candidate Generation of an Embedded/Induced Subtree	201
7.3.1 Counting Embedded k-subtrees	201
7.3.2 Counting Induced Subtrees by Enforcing Constraint	206
7.3.3 Counting Induced/Embedded Subtrees	209
7.4 Summary	211
<i>Chapter 8 Performance Evaluation of the Proposed Algorithms</i>	213
8.1 Introduction	213
8.2 The Rationale of the Experimental Comparison	213
8.3 Experimental Results & Discussions	215
8.3.1 Experiment Set I	217
8.3.1.1 Scalability Test	217
8.3.1.2 Pseudo-frequent Test	218
8.3.1.3 Deep Tree vs Wide Tree Test	219
8.3.1.4 Mixed (Deep and Wide) Test	220
8.3.1.5 Prions Test	221
8.3.1.6 CSLogs Test	222
8.3.1.7 Transaction-based Support Test	224
8.3.1.8 Overall Discussion	225
8.3.2 Experiment Set II	227
8.3.2.1 RMP Coordinate Test	228
8.3.2.2 Recursive List Test	228
8.3.2.3 Scalability Test	231
8.3.2.4 Deep Tree Test	231
8.3.2.5 Wide Tree	233
8.3.2.6 Constraining the Maximum Level of Embedding Test	234
8.3.2.7 Overall Discussion	236
8.3.3 Experiment Set III	236
8.3.3.1 Scalability Test	237
8.3.3.2 Frequent Subtrees over Different Support	237

8.3.3.3 Varying the Level of Embedding	239
8.3.3.4 Overall Discussion	240
8.4 Summary	241
References	245
<i>Chapter 9 Extension I: Mining Frequent Unordered Subtrees</i>	247
9.1 Introduction	247
9.2 The Needs for Mining Unordered Subtrees	249
9.3 The Problems and Characteristics of Frequent Unordered Subtrees Mining	250
9.4 UNI3 Miner: Mining Frequent Unordered Induced and Embedded Subtrees	252
9.4.1 Canonical Form Computation	253
9.4.2 Pre-conditions to Accelerate the Canonical Transformation	254
9.4.3 A Strategy for Mining Unordered Induced and Embedded Subtrees	254
9.4.4 UNI3 Miner Pseudo-code	256
9.5 Experimental Results & Discussions	257
9.5.1 Mining Induced Unordered Subtree Experiments	258
9.5.1.1 Time Performance Test	258
9.5.1.2 Scalability Test	259
9.5.1.3 Occurrence-match Support Test	260
9.5.2 Mining Embedded Unordered Subtree Experiments	260
9.5.2.1 Time Performance Test	261
9.5.2.2 Scalability Test	262
9.5.2.3 Ordered vs Unordered Test	263
9.5.3 Overall Conclusions	264
9.6 Summary	266
References	268
<i>Chapter 10 Extension II: Mining Frequent Subsequences</i>	271
10.1 Introduction	271
10.2 Overview of the Proposed Solutions	272
10.3 SEQUEST: Mining Frequent Subsequences from a Database of Sequences	275
10.3.1 Database Scanning	275
10.3.2 Constructing DMA-Strips	276

10.3.3 Enumeration of Subsequences	277
10.3.4 Frequency Counting	278
10.3.5 Pruning	279
10.4 SEQUEST Pseudo-code	280
10.5 Experimental Results & Discussions	280
10.5.1 Performance Test	281
10.5.2 Scalability Test	282
10.5.3 Frequency Distribution Test	283
10.5.4 Large Database Test	284
10.5.5 Overall Conclusions	284
10.6 Summary	285
References	287
<i>Chapter 11 Summary and Future Work</i>	289
11.1 Introduction	289
11.2 Recapitulation	289
11.3 Future Work	294
11.3.1 Distributed Parallel Processing	294
11.3.2 Mining Frequent Closed Subtrees	295
11.3.3 Mining Frequent Subgraphs	295
11.3.4 Enumeration of Canonical Form	296
11.3.5 Utilization of Hybrid Support	296
11.3.6 Application on Real-World Domains	297
11.4 Thesis Summary	297
References	299
<i>Appendix A -Implementation Issues</i>	300
A.1 Accelerating the Object Oriented (OO) Approach	300
A.2 Choosing Hash Functions	301
A.3 Hashing Integer Array versus String	302
A.3.1 Label Sensitivity Test	302
References	304

Table of Figures

Figure 3.1: Example of induced subtrees (T_1, T_2) and embedded subtrees (T_3) of tree T	75
Figure 3.2: Example of XML fragment	80
Figure 3.3: Illustration of element-element and element-attribute examples	81
Figure 3.4: Illustrations of tree-structured items (a) tree-structured items with size 3 (b) tree-structured items with size 1	82
Figure 3.5: Tree T_1, T_2, T_3 and T_4 with subtrees S_1 and S_2 to illustrate transaction-based and occurrence-match support definitions	86
Figure 3.6: Illustration of restricting the level of embedding	88
Figure 3.7: Example tree with labeled nodes ordered in preorder traversal	89
Figure 3.8: Snapshot of the representation of Human Prions protein data in XML format	90
Figure 3.9: Permutations of subtrees from an automorphism group $\text{Auto}(\{s_1, s_2, s_3, s_4\})$.	92
Figure 3.10: Examples illustrate the use of different support definition transaction-based and occurrence-match over ordered/unordered induced/embedded subtrees	95
Figure 4.1: A tree T to illustrate pseudo-frequent problem	118
Figure 5.1: Computing canonical form of tree T described in Chi, Yang & Muntz (2004a).	133
Figure 5.2: A fraction of XML tree	136
Figure 5.3: XML string index table	137
Figure 5.4: Integer-indexed tree of XML tree in Figure 5.2	137
Figure 5.5: An XML tree in Figure 5.2 formatted as a string-like representation as used in Zaki (2005b). tid: transaction-id; cid: omitted; S: size of the string	137
Figure 5.6: Illustration of generating a dictionary D from tree T where each cell in D has {label, level, scope, dpp} tuple.	140
Figure 5.7: A tree T with all of its 2-subtree candidates and the EL of tree T	142
Figure 5.8: $\text{VOL}('b \ c / e')$ of a subtree $\varphi: 'b \ c / e'$ when occurrence-match support is used	144
Figure 5.9: $\text{VOL}('b \ c / e')$ of a subtree with $\varphi: 'b \ c / e'$ when transaction-based support is used	144
Figure 5.10: An EL generated from tree T	146
Figure 5.11: The RL with line and arrow indicating the start and end of each list. Each cell denotes (pos, scope) of nodes of tree T	146
Figure 5.12: Illustration of the right-most-path coordinate (OC_{RMP}) of tree T_1, T_2 , and T_3	148
Figure 6.1: Diagram of a general mining frequent subtree framework (left) and the framework proposed in this thesis (right)	154
Figure 6.2: Example of induced subtrees (T_1, T_2, T_4, T_6) and embedded subtrees (T_3, T_5) of tree T	155
Figure 6.3: General structure of MB3 Miner	157
Figure 6.4: An arbitrary tree T	158

Figure 6.5: The dictionary structure of the tree T in Figure 6.4	158
Figure 6.6: EL and F_2 construction pseudo-code	159
Figure 6.7: Embedding list constructed from tree T in Figure 6.4	159
Figure 6.8: TMG enumeration: extending a subtree $T\{0,1,2,3,4,5\}$ with $OC_{RMP}:[0,3,4]$ with extension coordinates $[5, 6, 7, 8, 9]$	162
Figure 6.9: $VOL('a b c')$ of a subtree of tree T in Figure 6.8 with $\varphi:'a b c'$ when occurrence-match support is used	165
Figure 6.10: $VOL('a b c')$ of a subtree of tree T in Figure 6.8 with $\varphi:'a b c'$ when transaction-based support is used	165
Figure 6.11: MB3 Miner pseudo-code	168
Figure 6.12: Illustration of restricting the maximum level of embedding when generating $S1-4$ subtrees from a subtree $t_{k-1} \subseteq T$ whose encoding is $'a b'$ with $OC\{0,1\}$	170
Figure 6.13: iMB3 Miner pseudo-code	172
Figure 6.14: Examples of fictional family taxonomy of a great Professor for illustrating the importance of distance constraint	174
Figure 6.15: Example tree T with labeled nodes ordered in preorder traversal	176
Figure 6.16: Example of trees, tree $T1$ and tree $T2$	177
Figure 6.17: Illustration of extending $(k-1)$ -subtree T_k where $\varphi(T_{k-1}):'a b c / a // b c'$ and $OC_{RMP}(T_{k-1}): \{0,4,5\}$ with extension coordinates $\{6,7,8,9,10\}$	180
Figure 6.18: Frequency counting of a subtree t with encoding $'b e c / c'$ for which $t \subseteq T$	182
Figure 6.19: $[i]MB3^R$ pseudo-code	183
Figure 7.1: Example of an arbitrary tree $T1$ and its closed form $T2(3,2)$	191
Figure 7.2: Enumeration cost graph of uniform tree $T(3,2)$	197
Figure 7.3: Embedding list of a uniform tree $T(3,2)$	198
Figure 7.4: Enumeration cost graphs of enumerating $T(3,2)$	199
Figure 7.5: Illustration of the scope of multiplication between N_1 and N_2 of embedded subtrees (left) and induced subtrees (right)	207
Figure 7.6: A uniform tree $T(d,r)$ with height $d=2$ and $r=2$	209
Figure 7.7: Comparison of the numbers of embedded subtrees generated between $\delta:1$ (induced) and $\delta:2$ (embedded)	211
Figure 8.1: A table describing the characteristics of all the datasets used in experimental comparison throughout this thesis	216
Figure 8.2: Scalability test: (a) time performance (left) (b) number of subtrees $ C $ (right)	217
Figure 8.3: Pseudo-frequent test: number of frequent subtrees $ F $	218
Figure 8.4: A tree T to illustrate pseudo-frequent subtrees generation	218
Figure 8.5: Deep and wide tree test	220
Figure 8.6: Mixed dataset	220

Figure 8.7: Prions protein data: (a) time performance (left) (b) number of frequent subtrees (right)	221
Figure 8.8: Test on 54% transactions of original CSLogs data	223
Figure 8.9: Benchmarking the usage of transaction-based support for mining embedded subtrees	224
Figure 8.10: Benchmarking the usage of transaction-based support for mining induced subtrees	225
Figure 8.11: Testing RMP approach	228
Figure 8.12: Testing RL approach	230
Figure 8.13: Scalability test – time performance	231
Figure 8.14: Deep tree test for embedded subtrees	232
Figure 8.15: Deep tree test for induced subtrees	232
Figure 8.16: Wide tree test for embedded subtrees	233
Figure 8.17: Wide tree test for induced subtrees	233
Figure 8.18: Restricting the maximum level of embedding	235
Figure 8.19: Scalability test	237
Figure 8.20: Number of frequent subtrees detected for varying support thresholds	238
Figure 8.21: Varying maximum of level of embedding	239
Figure 9.1: Example of data where mining unordered subtree is more desirable than mining ordered subtree	249
Figure 9.2: Flow chart describing UNI3 Miner algorithm	252
Figure 9.3: UNI3 Miner algorithm pseudo-code	256
Figure 9.4: Time performance test	258
Figure 9.5: Scalability test	259
Figure 9.6: Occurrence-match support test	260
Figure 9.7: Performance test	261
Figure 9.8: Scalability test	262
Figure 9.9: Ordered vs unordered test	263
Figure 10.1: Example of a strip of a sequence $A \rightarrow CEF \rightarrow HJL$	276
Figure 10.2: Vertical tree representation of the strip in Figure 10.3	277
Figure 10.3: Enumeration examples	278
Figure 10.4: Pseudo-code of SEQUEST	280
Figure 10.5: 200K dataset (a) time performance (b) memory profiles	281
Figure 10.6: 100K, 500K, 1000K dataset (a) time performance (b) memory profiles	282
Figure 10.7: 100K, 500K, and 1000K dataset frequency distribution	283
Figure 10.8: 7000K time performance	284

Abstract

Association mining consists of two important problems, namely frequent patterns discovery and rule construction. The former task is considered to be a more challenging problem to solve. Because of its importance and application in a number of data mining tasks, it has become the focus of many studies. A substantial amount of research has gone into the development of efficient algorithms for mining patterns from large structured or relational data. Compared with the fruitful achievements in mining structured data, mining in the semi-structured world still remains at a preliminary stage. The most popular representative of the semi-structured data is XML. Mining frequent patterns from XML poses more challenges in comparison to mining frequent patterns from relational data because XML is a tree-structured data and has an ordered data context. Moreover, XML data in general is larger in data size due to richer contents and more meta-data. Dealing with XML, thus involves greater unprecedented complexity in comparison to mining relational data. Mining frequent patterns from XML can be recast as mining frequent tree structures from a database of XML documents. The increase of XML data and the need for mining semi-structured data has sparked a lot of interest in finding frequent rooted trees in forests.

In this thesis, we aim to develop a framework to mine frequent patterns from XML documents. The framework utilizes a structure-guided enumeration approach, *Tree Model Guided (TMG)*, for efficient enumeration of tree structure and it makes use of novel structures for fast enumeration and frequency counting. By utilizing a novel array-based structure, an *embedded list (EL)*, the framework offers a simple sequence-like tree enumeration technique. The effectiveness and extendibility of the framework is demonstrated in that it can be utilized not only for enumerating ordered subtrees but also for enumerating unordered subtrees and subsequences. Furthermore, the framework tackles the unprecedented complexity in mining frequent tree-structured patterns by generating only valid candidates with non-zero frequency count and employing a constraint-driven approach. Our experimental studies comparing the proposed framework with the state-of-the-art algorithms demonstrate the effectiveness and the efficiency of the proposed framework.